

Санкт-Петербургский государственный университет
Факультет прикладной математики–процессов управления
Кафедра управления медико-биологическими системами

Антропова Ксения Дмитриевна

Выпускная квалификационная работа бакалавра

**Оценка расстояния в задачах аппроксимации точечных
множеств многообразиями**

Направление 01.03.02

Прикладная математика и информатика

Заведующий кафедрой,
доктор физ.-мат. наук, профессор

Александров А.Ю.

Научный руководитель,
кандидат физ.-мат. наук, доцент

Гончарова М. В.

Рецензент,
кандидат физ.-мат. наук, доцент

Тамасян Г.Ш.

Санкт-Петербург,
2018 г.

Содержание

Введение	2
Обзор литературы	4
Глава 1. Анализ существующих подходов и алгоритмов . . .	6
1.1 Алгебраический подход	6
1.2 Геометрический подход	7
1.3 Алгоритм 3L	8
1.4 Алгоритм Левенберга-Марквардта	13
Глава 2. Предлагаемый подход	15
Глава 3. Экспериментальные результаты	17
Заключение	21
Список литературы	22

Введение

В задачах аппроксимации точечных множеств многообразиями важное место занимает подбор неявных полиномов. Неявные полиномы широко используются в прикладных исследованиях, например, в задачах компьютерного зрения, распознавания или моделирования объектов благодаря преимуществам перед другими способами представления данных в двумерном и трёхмерном пространствах. В частности, неявные полиномы не требуют параметризации и могут быть получены без предварительного анализа распределения данных, в отличие от сеток или В-сплайнов.

Обычно представление данных в виде неявных полиномов получают в результате процесса подбора (fitting process). Процесс подбора состоит в нахождении такой поверхности в трёхмерном пространстве и кривой в двумерном, которая будет наилучшим образом описывать заданный набор точек. Требуется найти наилучший неявный полином, который описывает заданный набор точек с помощью его корней. Другими словами, многочлен должен равняться нулю в заданных точках.

Пусть $f_c(X)$ – неявный многочлен степени d :

$$f_c(X) = \sum_{i+j+k \leq d} c_{i,j,k} \cdot x^i \cdot y^j \cdot z^k \quad (1)$$

Или в векторном виде:

$$f_c(X) = m^T c, \quad (2)$$

где $c = [c_{0,0,0}, c_{1,0,0}, \dots, c_{0,0,d}]^T$ – это вектор-столбец коэффициентов многочлена, у которого количество элементов равно $C_{d+3}^3 = \frac{(d+3)!}{d!3!}$; m – это вектор-столбец мономов, то есть $m = m(X) = [x^0 y^0 z^0, x^1 y^0 z^0, \dots, x^0 y^0 z^d]^T$.

Задача подбора состоит, во-первых, в выборе критерия близости множества корней найденного неявного полинома $Z_f = \{X : f_c(X) = 0\}$ к заданному множеству точек, а затем в минимизации этого критерия для нахождения лучшего вектора коэффициентов c .

Пусть $P = \Gamma_0 = \{p_i\}_1^N$ - множество заданных точек, лежащих в R^2 или в R^3 , тогда формально задачу подбора можно записать в виде:

$$\hat{c} = \arg \min_c Dist(P, f_c), \quad (3)$$

где $\arg \min_c$ обозначает вектор c коэффициентов полинома, при которых выражение $Dist$ (критерий близости, упомянутый выше) достигает минимального значения.

Для нахождения наилучшего вектора коэффициентов \hat{c} можно использовать алгебраический или геометрический подход. Важную роль играет выбор критерия близости $Dist(P, f_c)$, который часто основан на применении формул оценки расстояния от точки до кривой или поверхности. Задача данной работы состоит в применении новой формулы нахождения приближенного расстояния от точки до эллипсоида к задаче подбора неявных полиномов.

Обзор литературы

Одним из существующих методов подборки, то есть нахождения кривой в двумерном пространстве и поверхности в трёхмерном, которые будут наилучшим образом описывать данные, является интерполяция [1]. К достоинствам методов интерполяции относят единственность интерполяционного многочлена, однозначно определяемого набором заданных точек, а также возможность гарантировать то, что многочлен пройдёт через все эти заданные точки. Но здесь же скрыты и недостатки этих методов применительно к работе с прикладными задачами: если мы имеем набор из $n + 1$ точки, то степень интерполяционной кривой или поверхности равна n , и при нахождении этого объекта может возникнуть проблема решения большого числа уравнений. Кроме того, добавление новой точки приводит к изменению степени многочлена, из-за чего приходится пересчитывать заново коэффициенты.

Существует также метод сплайнов. Он заключается в формировании составной кривой из отдельных сегментов. Такой метод позволяет упростить расчёты, использовать уравнения невысоких степеней. Степень многочленов не зависит от количества точек. Существует также и много других методов интерполяции.

Одним из известных методов аппроксимации является метод наименьших квадратов. В применении к решаемой задаче метод наименьших квадратов (МНК) заключается в требовании построить такую кривую или поверхность, чтобы сумма квадратов отклонений построенного полинома от заданных точек была минимальной [2]. Существуют как геометрические подходы к аппроксимации, так и алгебраические. Нахождение первоначального полинома может осуществляться путём решения системы уравнений. Но существуют алгоритмы, например, алгоритм 3L [3], благодаря которым можно найти полином, который в последствии можно аппроксимировать с помощью метода наименьших квадратов для минимизации критерия качества подбора. Алгоритм 3L можно использовать в дальнейшем и для оптимизации критерия качества подбора, а можно проводить опти-

мизацию с помощью других алгоритмов, например, алгоритма Левенберга-Марквардта [4]. Алгоритм Левенберга-Марквардта хоть и является сложным в реализации, но он имеет высокую скорость сходимости и хорошо работает с задачей аппроксимации данных с помощью кривых или поверхностей.

Глава 1. Анализ существующих подходов и алгоритмов

1.1 Алгебраический подход [5]

Для поверхности, заданной с помощью неявного полинома в виде (2), справедливо утверждение: точка X лежит на поверхности тогда и только тогда, когда $f_c(X) = 0$. Следовательно, в качестве критерия оптимизации можно взять величину, известную как алгебраическое расстояние:

$$Dist(P, f_c) = \sum_{i=1}^N f_c^2(p_i) \quad (4)$$

В таком случае задача имеет очевидное решение $c = \mathbf{0}$, которое нам не подходит. Для того, чтобы исключить тривиальный вариант, накладывают нормирующее ограничение. Наиболее простым и удобным для использования нормирующим ограничением является выбор значения, равного 1, для одного из коэффициентов, например, $c_0 = 1$. Тогда получившаяся задача минимизации эквивалентна переопределённой системе уравнений

$$Mc = b, \quad (5)$$

где M - это матрица одночленов (каждый ряд содержит вектор одночленов $m^T(p_i)$, вычисленных в данной точке), $b = -\mathbf{1}$ - это вектор-столбец, содержащий -1 на каждой позиции, тогда оптимальное решение может быть алгебраически вычислено с помощью метода наименьших квадратов:

$$c = (M^T M)^{-1} M^T b. \quad (6)$$

Поскольку алгебраические подходы не являются итерационными, то их удобно использовать в приложениях. Однако, такие подходы обладают существенными недостатками: во-первых, неустойчивость множества решений и, во-вторых, недостаточность геометрической информации о данных. Избежать указанных проблем можно используя алгоритм 3L [3].

Алгоритм 3L представляет собой линейную полиномиальную подгонку методом наименьших квадратов, состоящую в генерации двух дополнительных наборов уровней, а именно $\Gamma_{+\delta}$ и $\Gamma_{-\delta}$ из заданного набора данных Γ_0 . Эти два дополнительных набора данных генерируются так, что один является внутренним, а другой внешним, и они размещаются на расстоянии $\pm\delta$ от исходных данных вдоль направления нормали. Следовательно, алгоритм 3L включает локальную геометрическую информацию, приводящую к более устойчивому решению. С учётом этих трёх наборов уровней, т. е. $\{\Gamma_{-\delta}, \Gamma_0, \Gamma_{+\delta}\}$, система (5) может быть переписана с использованием блок-матрицы M_{3L} и блочного вектор-столбца b в виде

$$M_{3L} = \begin{bmatrix} M_{\Gamma_{-\delta}} \\ M_{\Gamma_0} \\ M_{\Gamma_{+\delta}} \end{bmatrix} \quad b = \begin{bmatrix} -\epsilon \\ \mathbf{0} \\ +\epsilon \end{bmatrix}, \quad (7)$$

где M_{Γ_0} , $M_{\Gamma_{+\delta}}$, $M_{\Gamma_{-\delta}}$ – матрицы одночленов, вычисленные на исходном, внутреннем и внешнем множествах соответственно, а $\pm\epsilon$ – соответствующие ожидаемые значения во внутренних и внешних наборах уровней. Метрика расстояния, используемая в алгоритме 3L, имеет вид:

$$Dist(P, f_c) = \sum_{x \in \Gamma_0} f_c(x)^2 + \sum_{x \in \Gamma_{+\delta}} (f_c(x) - \epsilon)^2 + \sum_{x \in \Gamma_{-\delta}} (f_c(x) + \epsilon)^2. \quad (8)$$

Тогда решение МНК даёт для c :

$$c = (M_{3L}^T M_{3L})^{-1} M_{3L}^T b. \quad (9)$$

Чтобы повысить точность алгоритма 3L, был предложен алгоритм [6], являющийся алгебраическим подходом, который ослабляет дополнительные ограничения (7) так, что значения $\pm\epsilon$ независимо настраиваются для каждой отдельной точки.

1.2 Геометрический подход [5]

В геометрическом подходе расстояние между точкой и поверхностью обычно определяется как кратчайшее расстояние между этой точкой и

соответствующей ей точкой на поверхности (т. е. ортогональное расстояние [7]). Таким образом, в общем случае для геометрических методов мы имеем следующую оптимизационную задачу:

$$Dist(P, f_c) = \sum_{i=1}^N \min_{\hat{p}_i} d^2(p_i, \hat{p}_i), \quad (10)$$

где \hat{p}_i - точка на поверхности, соответствующая точке p_i . Теоретически, одновременно должны быть найдены как неизвестные параметры поверхности, так и точки \hat{p}_i . Но на практике эта задача решается следующим образом. Определяем некоторую начальную поверхность, а затем с помощью методов подбора улучшаем её до достижения сходимости. Таким образом, задача подбора разделяется на две стадии:

- 1) поиск точек \hat{p}_i ;
- 2) улучшение параметров поверхности.

1.3 Алгоритм 3L [3]

Рассмотрим для простоты данный алгоритм на примере двумерного случая. Пусть $f_c(X)$ – полином степени d :

$$f_c(X) = \sum_{\substack{0 \leq i, j \\ 0 \leq i+j \leq d}} c_{ij} x^i y^j = c_{00} + c_{10}x + c_{01}y + c_{20}x^2 + c_{11}xy + \dots + c_{0d}y^d$$

Представление неявной полиномиальной кривой для $f_c(X)$ задаётся формулой:

$$Z_f = \{X : f(X) = 0\}, \quad (11)$$

где Z_f – множество всех точек X , в которых полином $f_c(X)$ равен 0, и называется нулевым множеством для $f_c(X)$. Рассмотрим полином 4й степени от (x, y) :

$$\begin{aligned} f_c(X) = & c_{00} + c_{10}x + c_{01}y + c_{20}x^2 + c_{11}xy + c_{02}y^2 + \\ & + c_{30}x^3 + c_{21}x^2y + c_{12}xy^2 + c_{03}y^3 + c_{40}x^4 + c_{31}x^3y + \\ & + c_{22}x^2y^2 + c_{13}xy^3 + c_{04}y^4 \end{aligned} \quad (12)$$

или в векторной форме:

$$f_c(X) = m^T c, \quad (13)$$

где m – вектор-столбец (15×1) одночленов $m = [1 \ x \ y \ x^2 \ xy \ y^2 \ x^3 \ x^2y \ xy^2 \ y^3 \ x^4 \ x^3y \ x^2y^2 \ xy^3 \ y^4]^T$, c – вектор-столбец (15×1) коэффициентов полинома: $c = [c_{00} \ c_{10} \ c_{01} \ c_{20} \ c_{11} \ c_{02} \ c_{30} \ c_{21} \ c_{12} \ c_{03} \ c_{40} \ c_{31} \ c_{22} \ c_{13} \ c_{04}]^T$. Обозначим через Γ_0 множество точек данных X , которые в совокупности представляют объект 2-D. Как правило, Γ_0 состоит из точек вдоль границы объекта. Цель состоит в том, чтобы аппроксимировать Γ_0 нулевым множеством Z_f полинома четвёртой степени $f_c(X)$. Это может быть достигнуто путём минимизации функции ошибки

$$E = \sum_{X \in \Gamma_0} f_c^2(X) \quad (14)$$

или в векторной форме:

$$E = c^T M_{\Gamma_0}^T M_{\Gamma_0} c, \quad (15)$$

где i -ая строка M_{Γ_0} – это вектор m_i^T одночленов, вычисленных в точке (x_i, y_i) . Заметим, что $M_{\Gamma_0}^T M_{\Gamma_0}$ является положительной полуопределённой матрицей и $E \geq 0$. Так как величина E пропорциональна $\|c\|^2$ (квадрату длины c), то чтобы получить значащее значение c , минимизирующее (15), зафиксируем c единичной длины. Затем (15) минимизируется путём нахождения соответствующего направления для c . Соответствующим направлением для c является собственный вектор $M_{\Gamma_0}^T M_{\Gamma_0}$, отвечающий наименьшему собственному числу.

На рисунке 1 (а) показаны двумерные объекты, выбранные в качестве примера: круг, квадрат, треугольник, бабочка, самолёт, ботинок и гитара. Применяя только что описанную процедуру подбора собственного вектора, получаются полиномы, нулевые множества которых показаны на рисунке 1 (б). На рисунке 1 (с) показаны нулевые множества полиномов, полученных с использованием метода приближённого среднего квадратичного евклидова расстояния [8]. На рисунке 1 (д) представлены нулевые множества полиномов, полученных в результате применения 3L алгоритма.

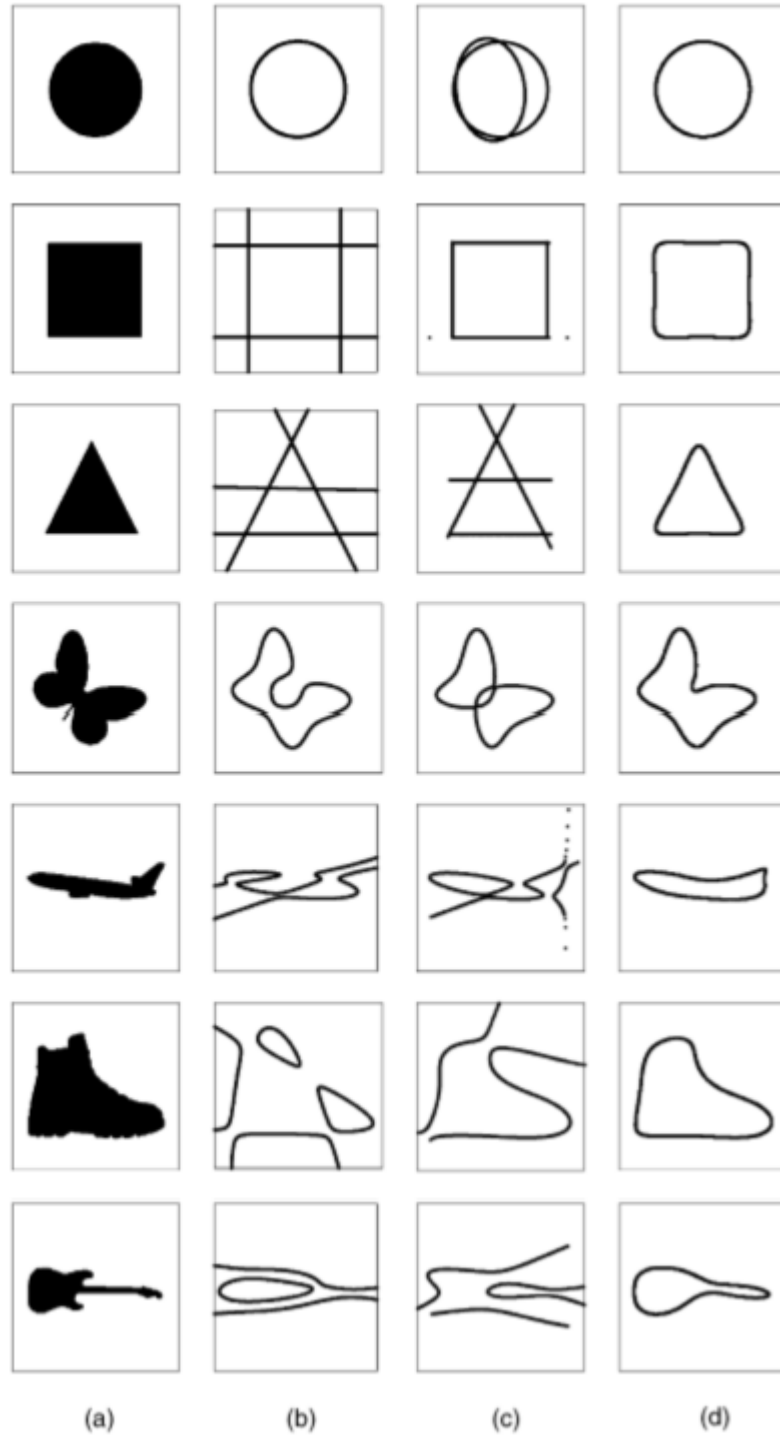


Рис. 1. [3] (a) Примеры объектов. (b) Метод собственного вектора. (c) [8]. (d) 3L алгоритм

Как видно из рисунка 1, 3L алгоритм наиболее точно из представленных для сравнения алгоритмов описывает заданные наборы точек. Для двух других алгоритмов нулевые наборы дают плохие представления Γ_0 .

Неудовлетворительные подгонки являются результатом минимизации конкретной выбранной функции (15). Небольшое количество шума в данных может привести к большим изменениям в векторе коэффициентов s и в соответствующей неявной полиномиальной кривой.

Посылка для подхода 3L-подгонки проста. Типичные полиномы, возникающие в результате минимизации (15), довольно гладкие в окрестности представленных данных, и хотелось бы ограничить поиск вектора s полиномиальных коэффициентов множеством многочленов, которые не имеют экстремальных или седловых точек (т. е. особенностей) вблизи Γ_0 . Алгоритм 3L достигает этого, создавая «ленточную поверхность» $\varphi(x, y)$ в окрестности Γ_0 и затем подбирая уже явный полином для поверхности $\varphi(x, y)$. Поверхность, которую мы используем, является монотонной, гладкой и обладает тем свойством, что вдоль любой нормали к локально гладкой аппроксимации Γ_0 функция $\varphi(x, y)$ положительна и возрастает при перемещении внутрь объекта и отрицательна и уменьшается при удалении от объекта. В результате полиномиальная поверхность, полученная подбором из $\varphi(x, y)$, проявляет такие же свойства. В частности, многочлен не имеет особых точек в окрестности данных и обычно положителен внутри объекта и отрицателен вне объекта вблизи ленты.

Конечно, такое ограничение полиномиальной подгонки уменьшает сложность двумерных и трёхмерных фигур, которые могут быть представлены, по причине меньшего количества степеней свободы, доступных для представления Γ_0 . В результате значение ошибки критерия близости больше, но результаты подгонки намного более стабильны.

В качестве функции $\varphi(x, y)$ удобно использовать подмножества так называемого евклидова расстояния со знаком в окрестности Γ_0 [9]. То есть значение $\varphi(x, y)$ является евклидовым расстоянием со знаком от точки (x, y) до ближайшей точки в наборе данных Γ_0 , и оно положительно для всех точек (x, y) внутри объекта и отрицательно для всех точек вне. Наша ленточная поверхность состоит из трех эквидистант: Γ_0 - точки, описывающие данные, $\Gamma_{+\delta}$ и $\Gamma_{-\delta}$. Здесь $\Gamma_{\pm\delta}$ - множества точек (x, y) , находящихся на расстоянии $\pm\delta$ от Γ_0 . На рисунке 2 изображены эти эквидистантные уров-

ни для двух примитивных двумерных объектов. Именно из использования трёх эквидистант (или уровней) выведено имя «3L алгоритм». При фактическом выполнении 3L-подбора объект сначала вписывают в единичный круг или сферу, центрированные в начале координат. Расчётное значение δ можно определять как процент от размера объекта, но на практике обычно берут фиксированное значение δ . Причём любое значение в диапазоне от 2 до 4 процентов даёт хорошие результаты.

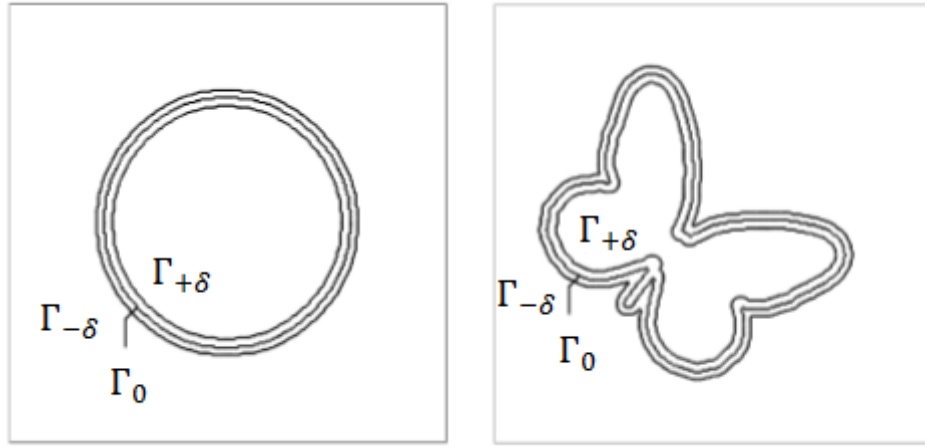


Рис. 2. [3] 3 набора уровней для 2-D объекта

Пусть $N_{-\delta}$, N_0 и $N_{+\delta}$ – количество точек (x, y) в $\Gamma_{-\delta}$, Γ_0 и $\Gamma_{+\delta}$ соответственно. Блочная матрица M_{3L} и вектор-столбец b имеют вид (7), M_{Γ_0} , $M_{\Gamma_{+\delta}}$, $M_{\Gamma_{-\delta}}$ – матрицы одночленов для соответствующих наборов точек на поверхности ленты $\varphi(x, y)$. Размерность матрицы M_{3L} и вектор-столбца b зависит от $N_{-\delta}$, N_0 и $N_{+\delta}$.

Тогда значение искомого набора коэффициентов c методом наименьших квадратов будет:

$$c = (M_{3L}^T M_{3L})^{-1} M_{3L}^T b. \quad (16)$$

Описанный метод является вычислительно простым, быстрым и удобным в применении.

Для того, чтобы обобщить 2-D случай до 3-D, достаточно просто переопределить M_{3L} и c , причем Γ_0 и $\Gamma_{\pm\delta}$ становятся поверхностями вместо

кривых. В частности, для полинома четвертой степени $f(x, y, z)$ m и c записываются в виде:

$$m = [1 \ x \ y \ z \ x^2 \ xy \ xz \ yz \ y^2 \ z^2 \ x^3 \ \dots]^T, \quad (17)$$

$$c = [c_{000} \ c_{100} \ c_{010} \ c_{001} \ c_{200} \ c_{110} \ c_{101} \ c_{011} \ c_{020} \ c_{002} \ c_{300} \ \dots]^T, \quad (18)$$

и метод (16) нахождения вектора c полиномиальных коэффициентов остаётся точно таким же, как для двумерного случая.

Таким образом, единственная проблема в обобщении алгоритма 3L на 3-D – это вычисление трёхмерных эквидистантных поверхностей. Поскольку поверхностные данные чаще всего состоят из набора неорганизованных точек 3-D-данных, то можно генерировать Γ_0 и $\Gamma_{\pm\delta}$ следующим образом.

Для каждой точки (x, y) из Γ_0 и её соседних точек, находящихся в окрестности с заданным радиусом, подбирают плоскость. Эту плоскость далее рассматривают как приближение касательной плоскости к поверхности в точке (x, y) . Далее, в направлении нормали к найденной плоскости на расстоянии $\pm\delta$ от неё получают точки для $\Gamma_{+\delta}$ и $\Gamma_{-\delta}$.

Сгенерированное множество точек остаётся только правильно соотнести с $\Gamma_{+\delta}$ и $\Gamma_{-\delta}$, это можно сделать, основываясь на некоторой предварительной информации об интересующем объекте.

1.4 Алгоритм Левенберга-Марквардта [4]

Алгоритм Левенберга-Марквардта является нелинейным методом наименьших квадратов, то есть минимизирует функцию

$$\sum_{i=0}^n f^2(x) \rightarrow \min. \quad (19)$$

Алгоритм является итеративным, если x_0 – это начальное приближение, то следующие приближения находятся по формуле

$$x_{i+1} = x_i - (H + \lambda I)^{-1} \nabla f(x_i), \quad (20)$$

где H – матрица Гессе для функции f , λ – параметр регулизации, который задаёт, как функция будет себя вести: как градиентный метод или метод

Ньютона. При маленьких λ будет плохая сходимость в окрестности точки, так как это проблема градиентного метода, а при больших λ нельзя гарантировать существование предела.

Глава 2. Предлагаемый подход

По примеру, предложенному в изучаемой литературе, был разработан следующий подход к подгонке неявных полиномов. Сначала с помощью 3L алгоритма строится набор данных Γ_0 , и вычисляются в зависимости от заданного значения δ два дополнительных набора $\Gamma_{+\delta}$ и $\Gamma_{-\delta}$. Далее решаем систему (16). На выходе получаем вектор коэффициентов полинома. Расстояние от каждой точки до кривой мы будем находить, используя следующую формулу:

$$d = d_s \left(1 + \frac{f_c}{2 \|grad f_c\|^4} k + \frac{f_c^2}{2 \|grad f_c\|^8} k^2 + \frac{5}{8 \|grad f_c\|^{12}} k^3 \right), \quad (21)$$

где $k = (grad f_c)^T H(f_c) grad f_c$; $H(f_c)$ - матрица Гессе, $d_s = \frac{|f_c|}{grad f_c}$.

Данная формула была выведена Утешевым А.Ю. в продолжение исследований [10]. Формула является новой и, можно сказать, только пробует для применения в прикладных задачах. В этом смысле работа обладает новизной. Как и некоторые другие конкурирующие подходы к нахождению приближённого расстояния, рассматриваемая формула хорошо работает не для всех точек относительно кривой или поверхности, поэтому некоторые точки в наборах данных пришлось убрать из рассмотрения.

В результате на предыдущем этапе было найдено расстояние от каждой отдельной точки до текущей кривой или поверхности. Совокупность всех этих расстояний обеспечивает хороший критерий качества для определения поверхности в форме наименьших квадратов:

$$Dist(P, f_c) = \sum_{i=1}^N d^2(p_i). \quad (22)$$

В данной реализации алгоритм Левенберга-Марквардта [11] используется для оптимизации расстояния (22) по параметрам кривой или поверхности. Алгоритм Левенберга-Марквардта начинается с исходного вектора

коэффициентов $c^0 = c$, полученного с помощью какой-либо алгебраической техники подгонки. В данной работе в качестве инициализации используется 3L алгоритм. Алгоритм Левенберга-Марквардта обновляет эти параметры итеративно следующим образом:

$$\begin{aligned} c^{t+1} &= c^t + \beta \Delta c \\ (J^T J + \lambda \text{diag}(J^T J)) \Delta c &= J^T D, \end{aligned} \tag{23}$$

где β – шаг восстановления; Δc представляет собой вектор коррекции параметров поверхности; λ – параметр затухания в алгоритме Левенберга-Марквардта; J – Якобиан вектора D ; а вектор $D = (d_1(c), \dots, d_n(c))^T$ соответствует расстояниям ($d_i(c) = d(p_i)$), норма l_2 которых должна быть минимизирована.

С помощью алгоритма Левенберга-Марквардта сначала вычисляется вектор расстояний D , затем вычисляется якобиан вектора D , из второго уравнения (23) находится вектор коррекции Δc и подставляется в первое уравнение для вычисления нового вектора c . Повторение итераций (23) происходит до наступления сходимости.

Итак, предложенный геометрический подход состоит из двух этапов. Во-первых, с помощью предлагаемого подхода оценивается остаточная ошибка от заданного набора точек до первоначальных коэффициентов неявного полинома. Затем коэффициенты неявного многочлена соответственно обновляются через алгоритм Левенберга-Марквардта. Два этапа повторяются до достижения желаемого результата.

Глава 3. Экспериментальные результаты

Рассматривался 3-D случай. Были взяты наборы точек¹, в которых количество точек варьировалось от 240 до 2000. Для программирования алгоритма был использован пакет прикладных программ Matlab.

Были загружены наборы точек и, используя 3L алгоритм [3], были построены 3 набора уровней: Γ_0 , $\Gamma_{+\delta}$ и $\Gamma_{-\delta}$. Во всех случаях значения точек данных центрированы и масштабируются между $[-1; 1]$. На рисунке 3 показаны 3 разных набора точек и проиллюстрированы по 3 набора уровней для каждого случая. Зелёным цветом обозначается уровень Γ_0 , красным цветом обозначается внешний уровень $\Gamma_{-\delta}$ и синим цветом – внутренний уровень $\Gamma_{+\delta}$. По наборам уровней Γ_0 , $\Gamma_{+\delta}$ и $\Gamma_{-\delta}$ были построены матрицы M_{Γ_0} , $M_{\Gamma_{+\delta}}$ и $M_{\Gamma_{-\delta}}$ соответственно, а с помощью этих матриц – блочная матрица M_{3L} . Далее, с помощью матрицы M_{3L} была решена система уравнений (16) и найден вектор коэффициентов c . С помощью данного вектора коэффициентов строится первоначальный полином, который далее оптимизировался для наилучшей “подгонки” к исходным точкам.

¹с сайта <http://shapes.aimatshape.net>

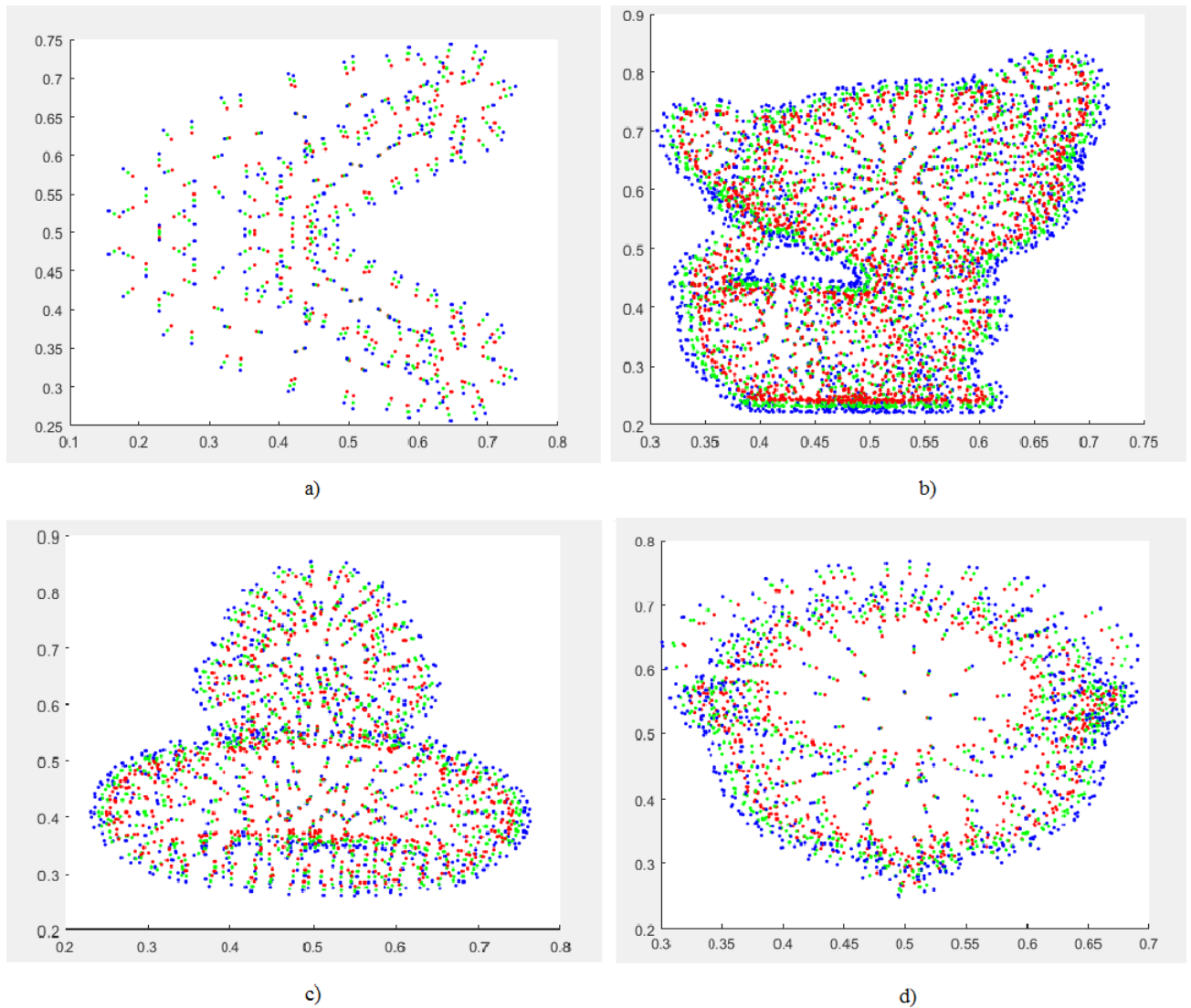


Рис. 3. Наборы точек и три набора уровней в трёхмерном пространстве: а) “узел”, б) “котёнок”, в) “утка”, г) “манекен”

Перед началом оптимизации было посчитано суммарное расстояние от точек до поверхности (21)-(22), чтобы потом сравнить этот критерий с новым значением после оптимизации вектора коэффициентов. Далее подбор осуществлялся итерационным методом Левенберга-Марквардта. Параметры инициализации для алгоритма 3L, параметры оптимизации для алгоритма Левенберга-Марквардта и критерий остановки эмпирически настроены, как представлено в таблице 1.

Таблица 1. Настройки параметров

Инициализация (3L)		Оптимизация (Л-М)		Критерий остановки
δ	ϵ	λ	β	число итераций
0.1	1	0.01	0.5	<25

Сравнение результатов накопленной ошибки (22) при разных количествах итераций представлено в таблице 2. Из таблицы 2 можно увидеть, из скольких точек состоит каждый набор точек и как быстро вектор коэффициентов оптимизируется для каждого набора точек.

Таблица 2. Значения критерия качества

	Сумма квадратов расстояний		
Количество точек	50 итераций	100 итераций	150 итераций
240 (“узел”)	101.2089	45.3162	18.5032
1557 (“котёнок”)	1.0220	0.9582	0.9205
745 (“утка”)	12.6904	10.7689	9.5468
639 (“манекен”)	0.0631	0.0622	0.0619

Также были проведены сравнения критериев качества других алгоритмов, результаты которых были взяты из [5]. Критерий остановки и параметры такие же, как в таблице 1. Так, в таблице 3 представлено сравнение накопленной ошибки для объектов “узел”, “котёнок”, “утка” и “манекен” с другими методами.

Таблица 3. Сравнение накопленной ошибки разных методов

	Степень полинома	3L алгоритм	Ортогональный фитинг	Предлагаемый подход
Рис. 4(а)	6	7.60	6.61	131.77
Рис. 4(б)	7	3.28	1.51	1.20
Рис. 4(с)	7	1.07	0.56	13.33
Рис. 4(д)	4	8.17	5.31	0.06

Из таблицы 3 видно, что предложенный метод работает в некоторых случаях лучше остальных, но в некоторых и хуже. В случае наборов точек

“котёнок” 4 (b) и “манекен” 4 (d) алгоритм работает гораздо лучше, чем ортогональный алгоритм и алгоритм 3L. Но в других двух случаях – “узел” 4 (a) и “утка” 4 (c) – данный метод сработал плохо, и критерий качества оказался слишком большим по сравнению с другими двумя методами. Тем не менее, предложенный в данной статье метод и формула для вычисления расстояния могут быть рассмотрены на одном уровне с другими алгоритмами, так как в половине случаев он работает лучше алгоритмов, результаты которых были представлены для сравнения. Дальнейшее исследование применимости предложенного подхода и выработка рекомендаций по его применению могут составить тематику моего магистерского исследования.

Для набора точек “манекен”, как для самого лучшего в плане критерия качества, была построена аппроксимированная поверхность, она представлена на рисунке 4. Из рисунка становится понятно, почему набору точек было дано название “манекен”.

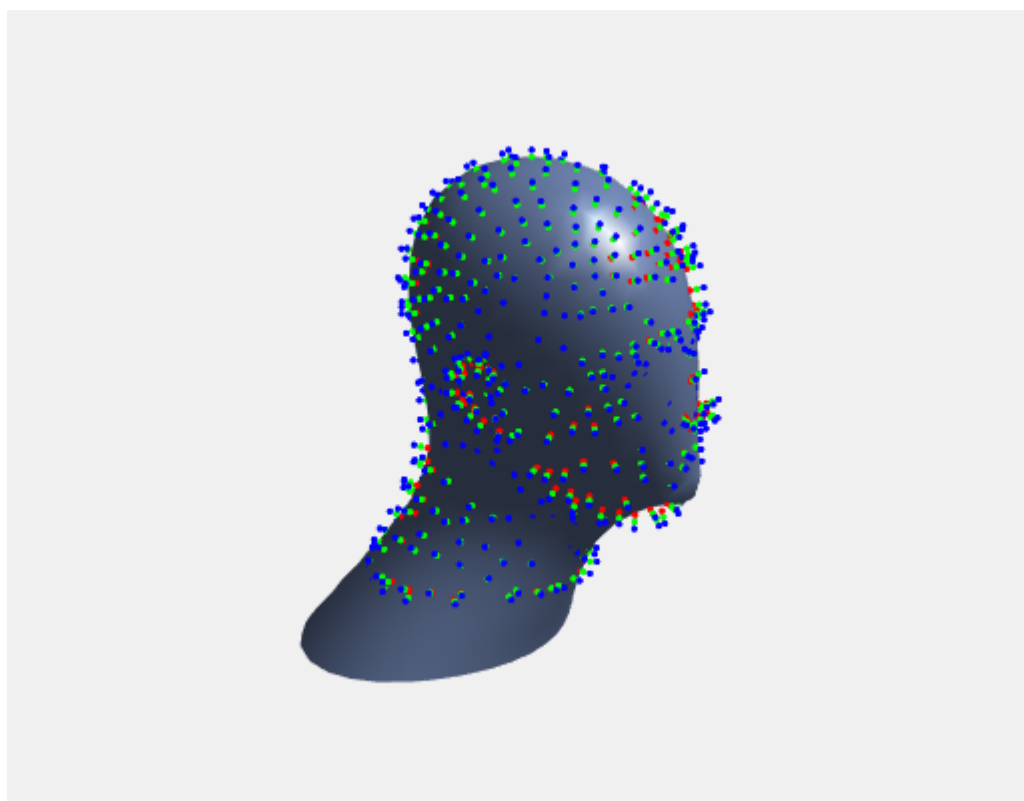


Рис. 4. Поверхность для набора точек “манекен” после оптимизации

Заключение

В работе выполнено сравнение новой формулы, полученной в результате развития исследований Утешева А.Ю. и Гончаровой М.В., с несколькими известными формулами. Основной упор сделан на работу в R^3 , так как для трёхмерного пространства, наиболее интересного с точки зрения приложений, известно ограниченное число методов нахождения приближённого расстояния. Была предложена методика сравнения с формированием тестовых данных и проведены количественные измерения по ряду критериев.

Список литературы

- [1] Шикин Е.В., Плис Л.И. Кривые и поверхности на экране компьютера. Руководство по сплайнам для пользователей. М.: ДИАЛОГ-МИФИ, 1996 г.
- [2] Зайдель А. Н. Ошибки измерений физических величин / А. Н. Зайдель. –М. : Наука, 1974.
- [3] M. Blane, Z. Lei, H. Civil, and D. Cooper The 3L algorithm for fitting implicit polynomials curves and surface to data. IEEE Trans. Pattern Anal. Mach. Intell., vol.22, no.3, pp.298–313, Mar.2000.
- [4] R. Fletcher, Practical Methods of Optimization, 2nd ed. New York: Wiley, 1990.
- [5] M. Rouhani, A. D. Sappa Implicit polynomial representation through a fast fitting error estimation. IEEE Transactions on image processing, vol.21, no.4, April 2012.
- [6] M. Rouhani and A. D. Sappa, Relaxing the 3L algorithm for an accurate implicit polynomial fitting. In Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog., San Francisco, CA, Jun. 2010, pp. 3066–3072.
- [7] S. Ahn, W. Rauh, H. Cho and H. Warnecke Orthogonal distance fitting of implicit curves and surfaces. IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 5, pp. 620–638, May 2002.
- [8] G. Taubin Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 13, no. 11, pp. 1,115-1,138, Nov. 1991.
- [9] P.E. Danielsson Euclidean distance mapping. Computer graphics and image processing, vol. 14, pp. 227-248, Nov. 1980.
- [10] Uteshev A.Yu., Goncharova M.V. Point-to-ellipse and point-to-ellipsoid distance equation analysis. J.Comput. Appl. Math., 2018, Vol. 328, P. 232-251.
- [11] Алгоритм Левенберга-Марквардта для нелинейного метода наимень-

ших квадратов и его реализация на Python [Электронный ресурс]//
Habr. – Режим доступа: <https://habr.com/post/308626/>, свободный.